**Defer**: defer schedules a function call to be run after the function completes. It is often used when resources need to be frees in some way.

For eg: package main

import "fmt"

func first() {

fmt.Println("1st")

}

func second() {

fmt.Println("2nd")

}

func main() {

defer second()

first()

}

Output: 1st

2nd

Basically defer moves the call to second to the end of the function:

func main() {

first()

second()

}

**Panic:** A *panic* generally indicates a programmer error (for example attempting to access an index of an array that's out of bounds, forgetting to initialize a map, etc.) or an exceptional condition that there's no easy way to recover from.

**Recover:** We can handle a run-time *panic* with the built-in recover function. *Recover* stops the *panic* and returns the value that was passed to the call to *panic.*

**How do you recover from a run-time panic?**

We can recover from a run-time *panic* by pairing it with *defer*

package main

import "fmt"

func main() {

defer func() {

str := recover()

fmt.Println(str)

}()

panic("PANIC")

}